TP11 – Restauration d'images

Débruitage

L'objectif du *débruitage* consiste, à partir d'une image bruitée $u_0 : \Omega \to \mathbb{R}$ (image en niveaux de gris), à trouver une image u telle que les deux images u et u_0 soient « proches », et que la nouvelle image u soit « lisse ». Le modèle variationnel de débruitage de type « Tikhonov » revient à minimiser l'énergie suivante :

$$E_{\text{Tikhonov}}(u) = \frac{1}{2} \iint_\Omega \left\{ [u(x,y) - u_0(x,y)]^2 + \lambda \, |\nabla u(x,y)|^2 \right\} dx\, dy \tag{1}$$

où $\lambda > 0$ est fixé par l'utilisateur. Le minimiseur de (1) est solution de l'équation d'Euler-Lagrange :

$$(I - \lambda \, \Delta)\, u(x,y) = u_0(x,y), \qquad \forall\, (x,y) \in \Omega \tag{2}$$

où I désigne l'opérateur identité et Δ le laplacien. Après discrétisation par différences finies, (2) se réécrit :

$$\underbrace{\left[\mathbf{I}_N - \lambda \left(-\mathbf{D}_x^\top \mathbf{D}_x - \mathbf{D}_y^\top \mathbf{D}_y\right)\right]}_{\mathbf{A}} \mathbf{u} = \underbrace{\mathbf{u}_0}_{\mathbf{b}} \tag{3}$$

où $\mathbf{u}$ et $\mathbf{u}_0$ sont des vecteurs de taille $N \times 1$ (N désigne le nombre de pixels) contenant les valeurs des niveaux de gris u et u_0, et $\mathbf{I}_N$ est la matrice identité (de taille $N \times N$). Enfin, $\mathbf{D}_x$ et $\mathbf{D}_y$ sont deux matrices bi-diagonales permettant d'approcher les composantes de ∇u par différences finies, dont la définition a été vue en cours.

Lancez `exercice_0`, qui implémente le modèle de débruitage (1) par résolution de l'équation discrète (3). Ce script utilise les fonctions `speye` pour construire la matrice identité, `spdiags` pour construire les matrices $\mathbf{D}_x$ et $\mathbf{D}_y$ (`sp` caractérise les fonctions pour matrices « creuses »), et l'opérateur \ (*backslash*) pour la résolution du système. Même en jouant sur le paramètre λ, vous constatez que ce modèle ne préserve pas convenablement les contours.

Exercice 1 : modèle de débruitage par variation totale

On préfère généralement à la régularisation quadratique $\frac{1}{2} \iint |\nabla u(x,y)|^2 \, dx\, dy$ du modèle (1), la « variation totale » $\iint |\nabla u(x,y)| \, dx\, dy$. L'énergie devient alors non différentiable, mais on peut utiliser l'approximation :

$$E_{\text{VT}}(u) = \iint_\Omega \left\{ \frac{1}{2} \, [u(x,y) - u_0(x,y)]^2 + \lambda \, \sqrt{|\nabla u(x,y)|^2 + \epsilon} \right\} dx\, dy \tag{4}$$

où le paramètre ϵ permet de garantir la différentiabilité en 0. L'équation d'Euler-Lagrange qui en résulte est non linéaire. Elle peut néanmoins être résolue par un schéma itératif de type « point fixe » :

$$\left[I - \lambda \, \nabla\cdot \left(\frac{1}{\sqrt{|\nabla u^{(k)}(x,y)|^2 + \epsilon}} \, \nabla \right) \right] u^{(k+1)}(x,y) = u_0(x,y), \qquad \forall\, (x,y) \in \Omega \tag{5}$$

où l'opérateur $\nabla\cdot$ désigne la *divergence*. Après discrétisation, l'itération (5) s'écrit :

$$\underbrace{\left[\mathbf{I}_N - \lambda \left(-\mathbf{D}_x^\top \mathbf{W}^{(k)} \mathbf{D}_x - \mathbf{D}_y^\top \mathbf{W}^{(k)} \mathbf{D}_y\right)\right]}_{\mathbf{A}^{(k)}} \mathbf{u}^{(k+1)} = \underbrace{\mathbf{u}_0}_{\mathbf{b}} \tag{6}$$

où $\mathbf{W}^{(k)}$ est la matrice diagonale des coefficients $\frac{1}{\sqrt{|\nabla u^{(k)}(x,y)|^2 + \epsilon}}$ (`spdiags` avec 0 en deuxième argument).

Complétez les parties manquantes (indiquées en commentaires) du script `exercice_1`, de manière à appliquer le modèle (4) à l'image `cameraman_avec_bruit.tif`, en effectuant les itérations de point fixe (6) avec $\epsilon = 0,01$, tant que $\|\mathbf{u}^{(k+1)} - \mathbf{u}^{(k)}\|_{L_2(\Omega)} > \|\mathbf{u}^{(k)}\|_{L_2(\Omega)}/1000$, en ajustant au mieux la valeur de λ dans l'intervalle $[10, 15]$. La commande `drawnow nocallbacks` permet de forcer l'affichage de la solution à chaque itération. La préservation des contours devrait être nettement meilleure. Faites ensuite une copie de ce script, de nom `exercice_1_bis`, afin d'étendre cette méthode à l'image `lena_avec_bruit.bmp`, en procédant canal par canal.

Inpainting

L'*inpainting* (qu'on pourrait traduire en français par « retouche d'images ») est une technique qui permet de restaurer une image sur un certain domaine. Il existe deux cas de figure assez différents, selon que le domaine à restaurer est déjà identifié ou non. Deux types d'applications sont possibles : la restauration de zones endommagées, par exemple les rayures sur des photographies anciennes ; la « réalité diminuée » (par opposition à la « réalité augmentée »), qui consiste à retirer certains objets d'une image. Nous nous intéressons dans l'exercice 2 à la restauration par *inpainting* (cf. figure 1). La réalité diminuée constituera le thème central du TP12.

FIGURE 1 – Restauration par *inpainting* : à gauche, image originale ; au centre, image dégradée u_0 (le domaine D à restaurer est indiqué en jaune) ; à droite, image u restaurée par *inpainting*.

Exercice 2 : modèle d'*inpainting* par variation totale

Le modèle d'*inpainting* par variation totale consiste en une modification très simple du modèle de débruitage (4). Supposons que la donnée u_0 ne soit pas fiable sur un domaine $D \subset \Omega$. Le terme d'attache aux données ne doit donc être calculé que sur $\Omega \backslash D$. Par conséquent, le modèle (4) devient :

$$E_{Inpainting}(u) = \frac{1}{2} \iint_{\Omega \backslash D} [u(x,y) - u_0(x,y)]^2 \, dx\, dy + \lambda \iint_{\Omega} \sqrt{|\nabla u(x,y)|^2 + \epsilon} \, dx\, dy \tag{7}$$

ce qui conduit aux itérations de point fixe suivantes :

$$\underbrace{\left[\mathbf{W}_{\Omega\backslash D} - \lambda \left(-\mathbf{D}_x^\top \mathbf{W}^{(k)} \mathbf{D}_x - \mathbf{D}_y^\top \mathbf{W}^{(k)} \mathbf{D}_y\right)\right]}_{\mathbf{A}^{(k)}} \mathbf{u}^{(k+1)} = \underbrace{\mathbf{W}_{\Omega\backslash D}\, \mathbf{u}_0}_{\mathbf{b}} \tag{8}$$

Dans cette expression, $\mathbf{W}_{\Omega\backslash D}$ est la matrice diagonale ayant pour éléments diagonaux les valeurs $1 - \chi_D(x,y)$, où χ_D désigne la fonction caractéristique de D : $\chi_D(x,y) = 1$ si $(x,y) \in D$, $\chi_D(x,y) = 0$ sinon.

Faites une copie du script `exercice_1_bis`, de nom `exercice_2`, que vous modifierez de manière à restaurer l'image `fleur_avec_defaut.png` (cf. figure 1), pour laquelle D est fourni dans l'image binaire `defaut_fleur.png`. Observez l'influence de λ, en testant différentes valeurs de ce paramètre, en particulier vis-à-vis du lissage (non forcément) en dehors du domaine D.

Une autre solution consiste à déterminer D par segmentation. Faites une copie du script `exercice_2`, de nom `exercice_2_bis`, que vous modifierez de manière à déterminer le domaine D en utilisant la couleur comme critère. Il est rappelé que le jaune n'est pas une couleur primaire, mais un mélange de rouge et de vert ne comportant pas de bleu. Testez ensuite ce script sur l'image `grenouille_avec_texte.png`, afin de « gommer » le texte affiché en surimpression.